Amendments to the Drawing:

Applicants have amended Figures 1-4 to include descriptive labels with the function of each box. Three replacement sheets are attached.

REMARKS

Claims 1-10 and 12 are pending in the above-referenced application and are submitted for the Examiner's reconsideration.

The Examiner objected to the drawings. In view of the amendments made thereto, withdrawal of this objection is respectfully requested.

Regarding the Section 101 rejection, the cancellation of claim 11 has rendered it moot.

Regarding the prior art rejection, the claimed invention enables a dynamical memory management without changing of the startup code. This is realized by partitioning of the memory and using a chained list. The startup code takes the information about the blocks which should be checked not from a list in the startup code itself, but from the blocks. So every block is easily to add to the system, because every block contains logistics information. So the memory is divided into logical blocks, whereby the purpose of the claimed invention is, that every function is connected to a logic block. For example the application program is connected to the application software block. By doing this, the partitioning of the memory in different blocks is achieved, and these blocks could be changed or exchanged individually.

Compared to this, Sorber shows a dividing of memory concerning different classes, whereby the memory parts of a class are connected by pointers. In the office action it is said, that a startup program is stored in the first memory block and it is said, that this is shown in column 7, paragraph 2. This is not true. In column 7, paragraph 2, it is shown, that the memory manager requests a large area of fixed non-swappable, available memory and receives a pointer or starting memory address from the operating system. So the startup is not from the first block of the class, but from the operating system. Compared to the claimed invention, this means that recited startup code should be received from outside, but the recited startup code takes the information by himself out of the blocks and not out of a list in the startup code itself, whereby the startup code is stored in the first memory block and against the opinion of the examiner this is not shown in the Sorber-document. Additionally, in column 7, lines 15 to 28, and column 8, lines 55 to 67, there is no sign of checking as in the claimed invention. Instead, in Sorber in the named lines it is only said, that the connected parts of the list are combined. So in the claimed invention, the going-through the chained list is checked and secured, which means, that the read information or data is checked, before they are evaluated and additionally as said in the office action, Sorbet does not teach. the

startup program obtains data for a check from the additional memory blocks as required by the claims.

This is also not shown in the Porterfield. There, especially in column 4, lines 3 to 14, it is only said, that a system address allocation table may specify the allocated addresses. So the addresses allocated for each computer device in the system address allocation table typically will be set by the basic input/output-system (BIOS)-software, when the computer system is initialized upon being turned on.

Therefore, it is not shown, that the going-through the chained list is secured and that, before an evaluation of the read information this read information is checked as in our invention.

It is therefore respectfully requested that the objections and rejections be withdrawn, and that the present application issue as early as possible.

Respectfully submitted,

KENYON & KENYON LLP

R: 42 (Reg. No. 41,172)

Dated: 2/10/06 By _____

Gerard A. Messina
(Reg. No. 35,952)

One Broadway
New York, New York 10004
(212) 425-7200